Proof: p_cˆsite(Zˆ2) >= 0.5567

This is a larger-window version of the comparison of J. van den Berg and A. Ermakov, "A new lower bound for the critical probability of site percolation on the square lattice", Random Structures Algorithms 8 (1996) 199-212 (below: "the paper"; its bound is 0.556). Labels: Sections 0-3 are PROVED-HERE, using the paper's Theorems 1.2 and 1.3 (KNOWN, cited). Section 4 is PROVED-HERE by exact computation (code/window.py), with a second implementation in code/independent/indep.py.

0. Setting (the paper, Sections 1-3; notation kept)

S is the square lattice Zˆ2. S* is its matching lattice: same vertices, with u ~ w iff ||u - w|| is 1 or sqrt 2. p_c(G) is the site-percolation critical probability of G, and theta_G(p) is the probability that a given vertex lies in an infinite open cluster (for L, take the max over the two vertices above a site).

KNOWN (the paper, Theorem 1.2, due to M. E. Fisher, J. Math. Phys. 2 (1961) 620-627): p_c(S) + p_c(S*) >= 1.

KNOWN (the paper, Theorem 1.3, going back to Strassen; the paper cites Liggett, Interacting Particle Systems, Chap. II, Thm 2.4): for laws mu, mu' on {0,1}ˆK, mu is stochastically dominated by mu' iff there is a coupling P of mu and mu' with P(w <= w') = 1.

The lattice L. Its vertex set is Zˆ2 x {0,1}, with two copies of S (layer 1 "upper", layer 0 "lower") and the vertical edge (v,0)-(v,1). For each edge {v, w} of S, let min be the endpoint with the smaller coordinate sum and max the other. There is also an edge from (min,1) to (max,0).

L is a subgraph of S*. Write a vertex of L as (x1, x2, x3), with (x1, x2) in Zˆ2 and x3 in {0,1}. The map phi(x1, x2, x3) = (x1 - x2, x1 + x2 + x3) is a bijection from Zˆ2 x {0,1} onto Zˆ2: given (a, b), we need x3 = (b - a) mod 2, then x1 + x2 = b - x3 and x1 - x2 = a. Under phi:

- in-layer edges become steps of (1,1) or (-1,1), which are diagonal edges of S*;
- the edge (v,0)-(v,1) becomes a step of (0,1), a vertical edge of S*;
- the edge from (v,1) to (v + e1, 0) becomes a step of (1,0), and the edge from (v,1) to (v + e2, 0) becomes a step of (-1,0); both are horizontal edges of S*. So phi maps L onto a subgraph of S*, namely S* with every other vertical edge removed. In particular p_c(S*) <= p_c(L).

Defect variables. For v in Zˆ2, eps(v) = (upper, lower) in {0,1}ˆ2 records whether (v,1) and (v,0) are open. The eps(v) are i.i.d., with each coordinate equal to 1 with probability p. Neighbours v, w of S are eps-adjacent iff an open vertex above min and an open vertex above max are joined by an edge of L. That is: upper(min) and upper(max), or lower(min) and lower(max), or upper(min) and lower(max). Write eta({v, w}) for this indicator; it is a function of (eps(v), eps(w)). Write pi for the law of one eps(v).

Lemma 0. If theta_S(1-p) <= C theta_L(p) for a constant C, then p_c(S) >= 1 - p. Proof. If theta_S(1-p) = 0, then 1 - p <= p_c(S), since theta_S(q) > 0 for every q > p_c(S). Otherwise theta_L(p) > 0, so p >= p_c(L) >= p_c(S*). By Fisher's inequality, p_c(S) >= 1 - p_c(S*) >= 1 - p. [] (This is the paper's Lemma 2.1, applied as the paper applies it.)

Growth process (the paper, Section 3). Fix a total order on Zˆ2 with least element O. Set xi(O) = 1 iff eps(O) is not (0,0). At each step, select the least vertex v with xi(v) = 1 that has a neighbour without a xi-value. Those neighbours are v's children, and each child u gets xi(u) = eta({u, v}). The process stops when no such v exists. Every vertex with xi = 1 is eps-connected to O.

1. Coupling lemma (the paper's "standard arguments", written out)

Lemma 1. Suppose that (i) P(xi(O) = 1) >= 1 - p, and (ii) at every step, for every history, the conditional law of the children's xi-values given the history stochastically dominates the product Bernoulli(1-p) law on the children. Then theta_S(1-p) <= 2 theta_L(p), and so, by Lemma 0, p_c(S) >=

1 - p.

Proof. Build zeta alongside xi, using auxiliary uniform random variables that are independent of eps.

- At the root, couple zeta(O) <= xi(O) with zeta(O) ~ Bernoulli(1-p).
- At each step with history h, let Q_h be a coupling, given by Theorem 1.3, of Bernoulli(1-p)^children and the conditional law of the children's xi-values, supported on {zeta <= xi}. Draw zeta on the children from Q_h(. | xi on the children) using fresh auxiliary uniforms.
- Vertices never assigned a xi-value get fresh independent Bernoulli(1-p) values.

zeta is i.i.d. Bernoulli(1-p). Earlier zeta values are functions of the earlier xi-values and of the auxiliary uniforms, and the uniforms are independent of eps. So the law of the new xi-values given (h, earlier zeta) is their law given h, and the new batch of zeta values is exactly Bernoulli(1-p)^children given everything before it. The number of children is a function of h. For the whole field, pre-sample independent Bernoulli(1-p) coins Y(v). After n steps, let the hybrid field be zeta on the vertices assigned so far and Y elsewhere. Each hybrid field is i.i.d. Bernoulli(1-p), by induction on n, and the hybrid fields converge pointwise to zeta. So every finite-dimensional law of zeta is the product law.

Comparison. Suppose the process dies out, after finitely many steps. Then no vertex with xi = 1 has a neighbour without a xi-value. So every neighbour u of a vertex with xi = 1 received its xi-value at some step, and there zeta(u) <= xi(u). Since zeta(O) <= xi(O), induction along any zeta-open path from O shows that every vertex of the path has xi = 1. So the zeta-open cluster of O lies in the finite set {xi = 1}.

Suppose instead the process survives. Then it has infinitely many steps, each selecting a distinct vertex with xi = 1, and all of these are eps-connected to O. That is an infinite eps-cluster, hence an infinite open cluster of L containing (O,0) or (O,1).

Therefore theta_S(1-p) = P(zeta-cluster of O is infinite) <= P(survival) <= 2 theta_L(p). []

(The case split is needed. For a general total order, a vertex with xi = 1 need not ever be selected when the process survives.)

2. Only the posterior of eps(b2) matters

Let b2 be the vertex selected at some step other than the first, and let c be its parent.

Lemma 2 (tree information). Let T be the exploration tree before this step: all vertices that have received a xi-value, each joined to its parent; vertices with xi = 0 are leaves. The history before this step is the event that eps(O) is open and eta(e) = x_e for every edge e of T, with the x_e as observed. Given the history, the law of (eps(v))*v is proportional to prod_v pi(eps(v)) * prod*{e in T} 1[eta(e) = x_e]. The children of b2 are vertices without a xi-value, so they are not in T, and their eps are independent of the history and of eps(T). Consequently, the conditional law of the children's xi-values given the history is sum_beta w(beta) nu_beta. Here w is the posterior law of eps(b2) given the history, and nu_beta is the product law of the children's eta given eps(b2) = beta.

Proof. The process is a deterministic function of the eta values on the edges of T revealed so far. Which vertex is selected, who its children are, and when the process stops depend only on which vertices have xi-values and what those values are, and xi(u) = eta({u, parent(u)}). So the history is exactly the stated event. Each eta(e) depends only on the eps of the two endpoints of e, and the eps are independent, which gives the displayed density. A child a of b2 has no xi-value, so no edge of T contains a, and eps(a) does not appear in the event. (The child may be adjacent in S to other vertices of T through edges that are not in T. Their eta values are not part of the history, so they do not matter.) Given eps(b2), the eta({a_i, b2}) are functions of eps(b2) and the independent eps(a_i). []

T is a tree: every vertex except O has exactly one parent, assigned when it received its xi-value. So w is computed exactly by message passing on T. Write A(gamma, beta) = 1[eta({c, b2}) = 1 when eps(c) =

gamma and eps(b2) = beta]. Then

```
w(beta) ∝ pi(beta) sum_gamma A(gamma, beta) mu_c(gamma),
```

where mu_c(gamma) ∝ pi(gamma) times the product of the messages into c from its tree neighbours other than b2: its parent side, and each of its other children.

3. The worst case is at the extreme rays (cone argument)

For an increasing event B of the children, domination at this step for B reads

```
sum_beta w(beta) (nu_beta(B) − pi_{1−p}(B)) >= 0,
```

with w unnormalized (the normaliser is positive). Write mu_c(gamma) = pi(gamma) M_par(gamma) prod_s M_s(gamma). Then the left side is multilinear in the messages (M_par, M_s, ...): each message appears once, linearly. Suppose each message ranges over a set contained in a cone K_par, K_s, ... generated by finitely many vectors, and the inequality holds whenever every message is one of the generators. Then it holds for all messages in the cones. To see this, fix all but one message: the left side is linear in the remaining one and nonnegative on its generators, hence on its cone. Then induct over the messages.

Orientation. Call a tree edge "max" if the child is the max neighbour of its parent (the larger coordinate sum), and "min" otherwise; A depends only on this orientation. A vertex whose parent is its min neighbour has child slots (max, max, min). A vertex whose parent is its max neighbour has (max, min, min). The root O has (max, max, min, min).

Messages into a vertex from a child slot of orientation o:

- all ones, if the slot is absent from T as a child of this vertex;
- the eta = 0 leaf message, sum_beta pi(beta) (1 - A_o(., beta)), with the sum over all four states;
- for eta = 1, sum_beta pi(beta) A_o(., beta) prod_slots M(beta) over the child's own slots. An unprocessed child has all of its slots absent.

The message from the parent side is sum_delta A_o(delta, .) pi(delta) M_par(delta) prod_{other slots} M(delta). Here M_par comes from the next level up, or is all ones if the parent is O (which has four slots).

Cones. D_k and U_k are the cones of these messages for subtrees of depth k. At depth 0, the subtree beyond the vertex is replaced by an arbitrary nonnegative function of its eps (the paper's adversarial point mass); the resulting messages lie in the cone generated by A_o(., beta) over open beta. Every real history's messages lie in these cones. This is a superset: geometric constraints, such as sites being distinct, are ignored, which can only make the check harder. Each multilinear image of a product of cones lies in the cone generated by the images of generator tuples. window.py keeps the extreme rays (an exact 2D convex hull in projective coordinates). It then encloses each extreme ray in a small grid triangle with vertices >= 0 summing to 1 (GRID = 2^40), which again gives a superset cone, with small denominators.

Root step. O has four children and no history beyond xi(O) = 1, so its domination is checked directly on all 166 increasing events of {0,1}^4 (code/validate.py, root_step): the margin at p = 0.4433 is 1.470119e-2. Also P(xi(O) = 1) = 1 - (1-p)^2 >= 1 - p.

Fewer than three children. The paper's argument applies unchanged: add imaginary children with fresh eps. The law of the real children is a marginal of the three-slot law, and marginals of a dominating law dominate.

4. The computation (exact rational arithmetic, Python fractions)

code/window.py checks the inequality of Section 3 for:

- all generator tuples;
- all 18 nontrivial increasing events of {0,1}^3 (the empty and full events hold trivially);
- b2 a max or a min child of c;
- c's parent side min or max, or c = O.

Results at p = 4433/10000 (outputs/cert_output.txt, result/bound.json):

depth	worst margin
0	-1.207390e-3 (fails; this is the paper's window)
1	-4.498328e-5 (fails)
2	+1.362252e-4
3	+1.658528e-4

Every case passes at depth 2 and at depth 3. With Sections 0-3: **p_c^site(Z^2) >= 1 - 4433/10000 = 0.5567.**

Cross-checks:

- **Depth 0 reproduces the paper:** the check holds exactly for p >= 0.443917087, and the worst margin at p = 0.444 is 1.621780e-4. This is the paper's A1 u A2 margin, 0.803026 - 0.802864.
- **code/vdbe1996.py** reproduces the paper's Proposition 4.2 and all 13 rows of its table on p. 210 (outputs/vdbe1996_output.txt).
- **code/independent/indep.py** is a second implementation that does not import window.py. It uses explicit directions in Z^2, no symmetry reduction, and exact cones with no rounding. It gives the same worst margins at depths 0-3 (outputs/indep_margins_output.txt).
- **code/validate.py** checks message passing against brute-force enumeration on random explicit trees embedded in Z^2, and checks the root step. It imports window.py, so it is a consistency check only.

Best bounds by depth from this relaxation (bisection to 1e-9): 0.556083, 0.556677, 0.556770, 0.556785, rising to 0.556788090 from depth 8. That is the limit of this cone relaxation, which ignores geometry. The claim uses only the depth-3 check at p = 4433/10000.

5. Where a reader should look hardest

1. Lemma 2: is anything other than the tree-edge eta values revealed, for example by the selection order?
2. The superset: every real history must map to one of the modelled cases (slot orientation; absent, leaf or processed; root).
3. The code: the hull, the outward rounding, the 18 increasing events, the orientation conventions.

Errata (corrections the panel asked for; the panelled files are left exactly as panelled)

None of these changes the result or the certificate.

1. the-record-we-beat and the claim's literature record describe arXiv:2405.07345 as "a bound for the truncated square lattice". It is T. Koehler-Schindler and M. Sulser, "Critical probabilities for positively associated, finite-range dependent percolation models". It uses a lower bound for site percolation on the truncated square lattice as an input. It is not a bound for Z^2 and does not compete (NOVELTY).
2. object/PROOF.md, Section 0: the total order should be of type omega, for example an enumeration of Z^2 starting at O, so that "the least vertex with ..." always exists. Any such enumeration works (REPRODUCE).
3. object/PROOF.md, Lemma 2: the displayed density omits the factor 1[eps(O) open]. That factor is implied at every step after the first, because O then has a tree edge with eta = 1 (REPRODUCE).
4. The root step is checked by how-we-built-it/code/validate.py, which prints its margin (1.470119e-2) but does not assert it is positive, and it is not recorded in object/bound.json. verify-it-yourself/check_bound.py now recomputes it from the definition and asserts it. Two skeptics confirmed it independently (CRITERION).
5. The claim's phrase "enlarged to the whole exploration-tree history" should read "exact to depth 3, with universal outer cones covering the rest of the history" (CRITERION).